Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1 to 13 (Cancelled)

14. (Currently amended) A method for granting threads running on various multi-processor nodes within a multi-computer system ownership of a global synchronization object comprising the steps of:

maintaining a record of the state of the global synchronization object as free, owned, or in transition;

when a thread seeks ownership of the global synchronization object, granting the thread, through a spinlock mechanism, access to the status of the global synchronization object, and granting the thread ownership if the global synchronization object is free;

if the global synchronization object is ~~not free~~ [[(]] owned or in transition [[)]] , adding the thread's node to a queue of nodes having threads awaiting ownership of the global synchronization object, and permitting the thread to seek ownership of a local synchronization object established on the thread's node by a local operating system, but temporarily blocking other threads on the thread's node from seeking ownership of the local synchronization object and forcing them into suspension;

when the global synchronization object ownership is released by a thread, lacing the global synchronization object into its transition state, and then arranging for each node in the queue, in turn, to stop blocking threads on its node from seeking ownership of the local synchronization object, and permitting any thread that then gains ownership of its local synchronization object to resume execution and to gain ownership of the global synchronization object if the object is ~~not owned~~ [[(]] free or in transition [[)]] , this process continuing until the global synchronization object is owned or until no more threads seek its ownership, at which point the global synchronization object enters its free state.

15. (Previously presented) A method for granting threads running on various multi-processor nodes within a multi-computer system ownership of a global synchronization object in accordance with claim 14 wherein the local or global synchronization objects are mutexes.

16. (Previously presented) A method for granting threads running on various multi-processor nodes within a multi-computer system ownership of a global synchronization object in accordance with claim 14 wherein the local or global synchronization objects are semaphores.

17. (Currently amended) A set of synchronization software computer programs designed for use in conjunction with a multi-computer system, where individual nodes have their own copies of an operating system with local node synchronization software included in the operating system, said synchronization software computer programs being capable of carrying out the following steps to implement global synchronization objects:

maintaining a record of the state of each global synchronization object as free, owned, or in transition;

when a thread seeks ownership of a global synchronization object, granting the thread, through a spinlock mechanism, access to the status of the global synchronization object, and granting the thread ownership if the global synchronization object is free;

if the global synchronization object is ~~not free~~ [[(]] owned or in transition [[)]] , adding the thread's node to a queue of nodes having threads awaiting ownership of the global synchronization object, and permitting the thread to seek ownership of a local synchronization object established on the thread's node by a local operating system, but temporarily blocking other threads on the thread's node from seeking ownership of the local synchronization object and forcing them into suspension;

when the global synchronization object ownership is released by a thread, placing the global synchronization object into its transition state, and then arranging for each node in the queue, in turn, to stop blocking threads on its node from seeking ownership of the local synchronization object, and permitting any thread that then gains ownership of its local synchronization object to resume execution and to gain ownership of the global synchronization object if the object is ~~not owned~~ [[(]] free or in transition [[)]] , this process continuing until the global synchronization object is owned or until no more threads seek its ownership, at which point the global synchronization object enters its free state.

18. (Previously presented) A set of synchronization software computer programs designed for use in conjunction with a multi-computer system in accordance with claim 17 wherein the local or global synchronization objects are mutexes.

19. (Previously presented) A set of synchronization software computer programs designed for use in conjunction with a multi-computer system in accordance with claim 17 wherein the local or global synchronization objects are semaphores.